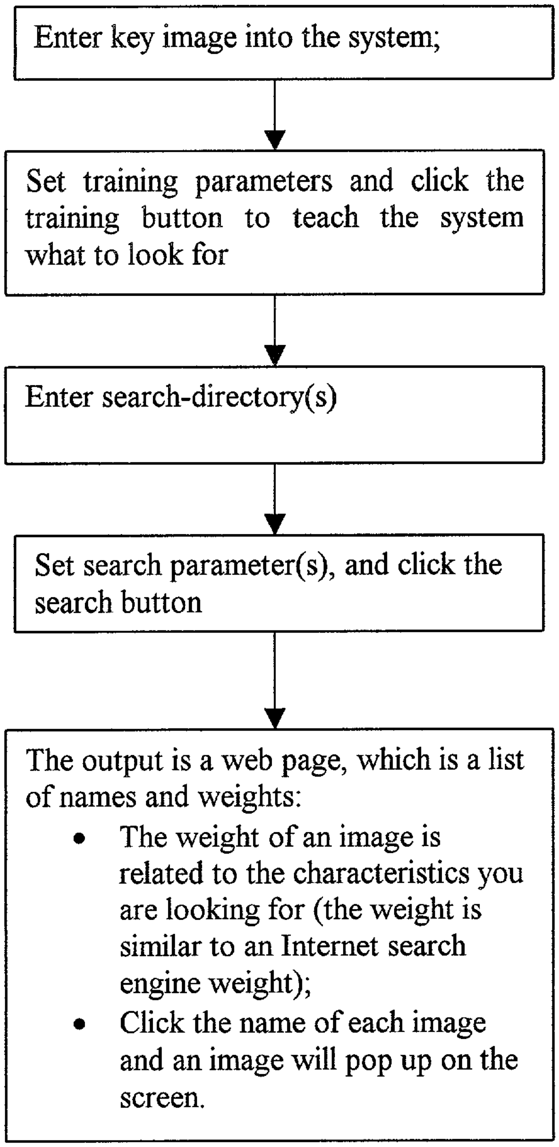

Figure 1. Search Process, which is applicable to image verification, identification, and retrieval.

1. Enter key image into the system;
2. Set training parameters and click the training button to teach the system what to look for;
3. Enter search-directory(s);
4. Set search parameter(s), and click the search button;
5. Repeat the above process for each class and then click the "Record" button. At the end, click the "Classification" button. The output web page will first list the sample images for each class. Then it will list:
   - An image link for each image in the search directory;
   - The classification weights of this image in each search; and
   - The classification of this image as a link.

Figure 2. Classification Process

1. Provide the batch code to the system, which includes:
   - Click the save button to save the current setting, including key(s), search directory(s), and parameters into a batch code.
   - Click a file button to recall one of the many batch codes saved earlier.
   - Cut and paste or simply type in a batch code by keyboard.
2. Click batch button to execute the code.

Figure 3. Batch Process

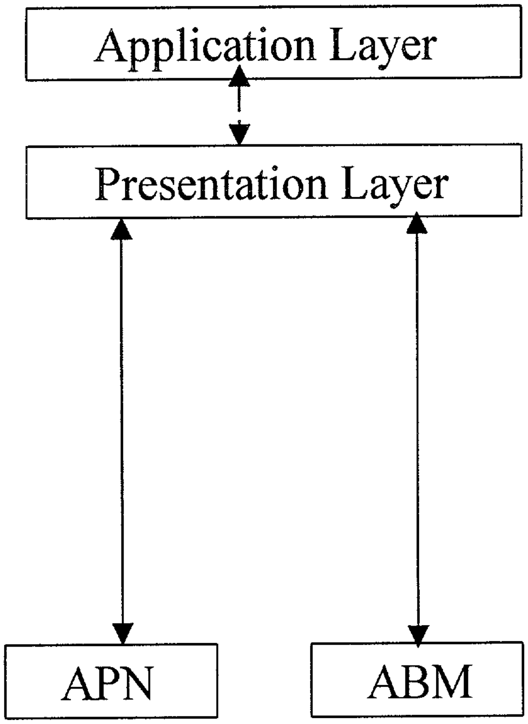

Figure 4. 3-layer Architecture

1. Create an ABM net with no connections;
2. Combine an image and its classification into an input vector.
3. Impose the input vector to the learning module.
4. The ABM neural connections are calculated based on the input vector. Let N be the number of neurons; the order of connections can be up to N and the number of connections can be 2**N, where ** represent the exponential function.
5. The Markov chain is formed after the connections are established. This Markov chain will settle on its invariant distribution. A distribution function is deployed to describe such a distribution.
6. This distribution function, once obtained, can be used to classify images. This will produce triplets of image, class, and weight. Image retrieval and classification are two different sides of the same token.
7. These triplets of image, classification, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

Figure 5. ABM Neural Layer Overview.

1. Create a ABM neural net with no connections;
2. Combine an image and its classification into an input vector.
3. Impose the input vector to the learning module.
4. The ABM neural connections are calculated based on the input vector. Let N is the number of neurons, these connections can be up to the order of N.
5. The Markov chain is formed after the connections are established. This chain will settle on its on its invariant distribution. A distribution function is deployed to describe such a distribution.
6. This distribution function, once obtained, can be used classify images. This will produce triplets of image, classification, and weight. Image retrieval and classification are two different sides of the same token.
7. These triplets of image, classification, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The classification is omitted because the search problem has only one class.

Figure 6. APN Neural Layer Overview.

1. Open file from the image source;
2. Decode the image into pixels arrays;
3. Process images with a filter;
4. Reduce the size of images to an internal representation. The users can arbitrarily choose the internal representation of the images. Such reduction can be based on individual images on a case-by-case reduction, or deploy the same reduction factor across to all images.
5. In the case where many pixels in an images have to combined into a new pixel before leaving this layer, the user can choose a reduction type such as taking average, maximum, minimum, or deploy a threshold.
6. Pass the image array to the next layer.

Figure 7. Presentation Layer Overview.

1. Delete the existing ABM connections;
2. Combine an image and its classification into an input vector.
3. The ABM neural connections are calculated based on the input vector. Let N is the number of neurons, these connections can be up to the order of N. The image is randomly breaking down into a predefined number of pieces.
4. Let an image piece, p1, have K = (k1 + k2) pixels, where K is an integer. After imposing the pixel vector to the ABM net, k1 is the number of neurons excited and k2 is the neurons of neurons grounded. A neural state vector can be constructed to represent such a configuration, which k1 components being 1 and k2 components being 0.
5. All such vectors together form a space, the connection space. A distance, either the Hausdorff distance or L1 distance or L2 distance can be defined in this space. Such a definition of a distance allows all possible connection vectors to be classified via a distance from p1. Many vectors will be in a group with distance 1 from p1. Many vectors will be in a group with distance 2 from p1, ...
6. The connection represented by p1 is assigned the largest synaptic connection weight. Those connections in the distance 1 group will have smaller weights, ... After a certain distance, the connection weights will be 0, or there will be no connections. The present invention covers all possible combinations of such a generating method.
7. The Markov chain is formed after the connections are established.

Figure 8. ABM Training Algorithm.

1. Delete the existing ABM connections;
2. Combine an image and its classification into an input vector.
3. The ABM neural connections are calculated based on the input vector. Let N is the number of neurons, these connections can be up to the order of N. The image is randomly breaking down into a predefined number of pieces.
4. Let an image piece, p1, have K = (k1 + k2) pixels, where K is an integer. After imposing the pixel vector to the ABM net, k1 is the number of neurons excited and k2 is the neurons of neurons grounded. A neural state vector can be constructed to represent such a configuration, which k1 components being 1 and k2 components being 0.
5. All such vectors together form a space, the connection space. A distance, either the Hausdorff distance or L1 distance or L2 distance can be defined in this space. Such a definition of a distance allows all possible connection vectors to be classified via a distance from p1. Many vectors will be in a group with distance 1 from p1. Many vectors will be in a group with distance 2 from p1, ...
6. The connection represented by p1 is assigned the largest synaptic connection weight. Those connections in the distance 1 group will have smaller weights, ... After a certain distance, the connection weights will be 0, or there will be no connections. The present invention covers all possible combinations of such a generating method.
7. The Markov chain is formed after the connections are established.
8. For each connection, in addition to the synaptic connection weight, a mapping over each connection is established. Let k1 be a number of neurons in the original k1 order connection generated by p1, then this mapping maps from the k1 neuron to the k1 pixel value which excited these neurons. This completes the connection for the original segment p1.
9. The segment, p1, also generated many other connections. If a neuron in this connection is one of the original k1 neurons in p1, then this neuron is mapped into the corresponding pixel value, which cause this neuron excited; otherwise, this neurons is mapped into 0. This completes the mappings of all connections generated by this segment p1.

Figure 9. APN Training Algorithm.

1. An image to be classified is imposed on the Markov Chain.
2. This Markov chain will settle on its invariant distribution. A distribution function is deployed to describe such a distribution.
3. This distribution function, once obtained, can be used to classify images. This will produce triplets of image, class, and weight. Image retrieval and classification are two different sides of the same token.
4. These triplets of image, classification, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

Figure 10. ABM Recognition Algorithm.

1. An image to be classified is imposed on the Markov Chain.
2. This chain will settle on its on its invariant distribution. A distribution function is deployed to describe such a distribution.
3. This distribution function, once obtained, can be used classify images. This will produce triplets of image, class, and weight.
4. Comparing the input-vector and the APN-connection-vector modifies this weight. All connection vectors together forms a vector space. A distance, either L1 distance or L2 distance can be defined in this space. The new weight will be directly proportional to the old weight and inversely proportional to this distance. This will produce a new set of triplets of image, classification, and weight.
5. These triplets of image, classification, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

Figure 11. ABM Recognition Algorithm.

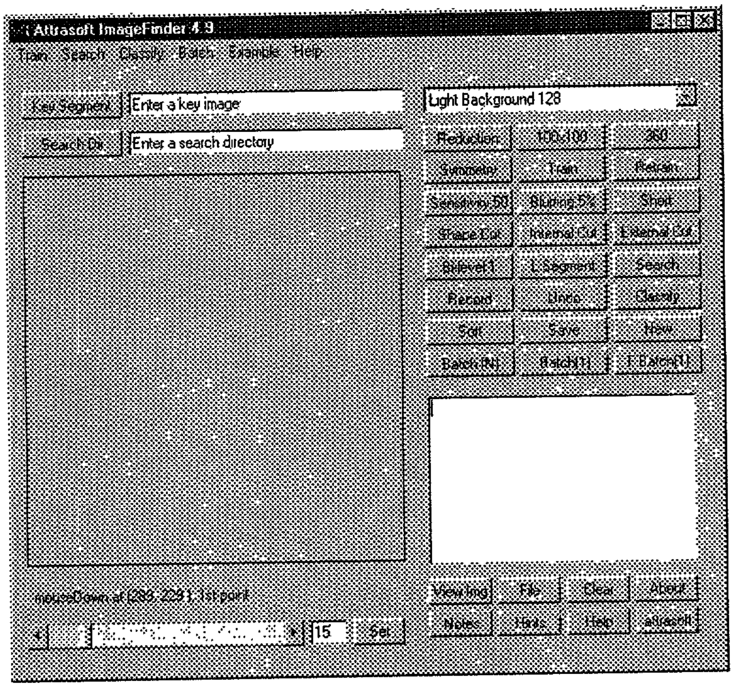

Figure 12. Sample User Interface of the Present Invention.

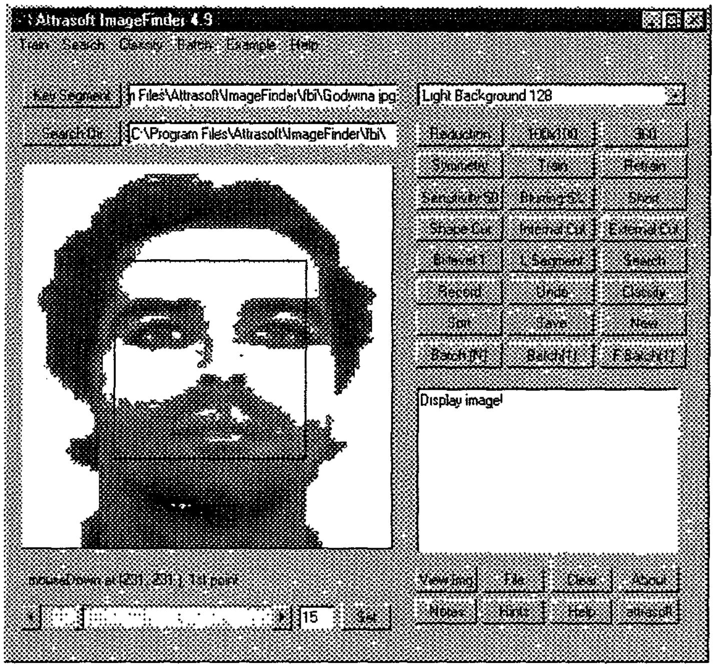

Figure 13. Sample Key Input for the Present Invention.

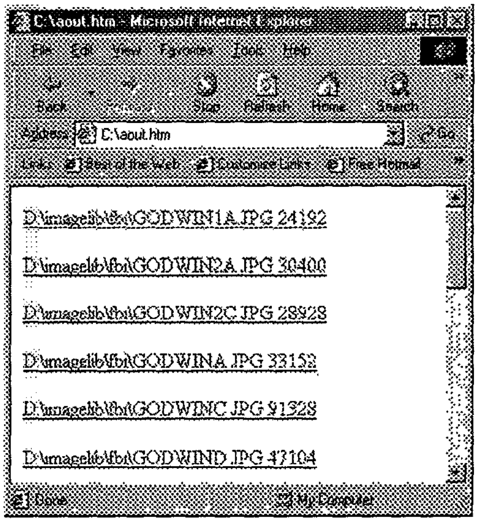

Figure 14. Sample Search Output of the Present Invention. The search output is a list of pairs.

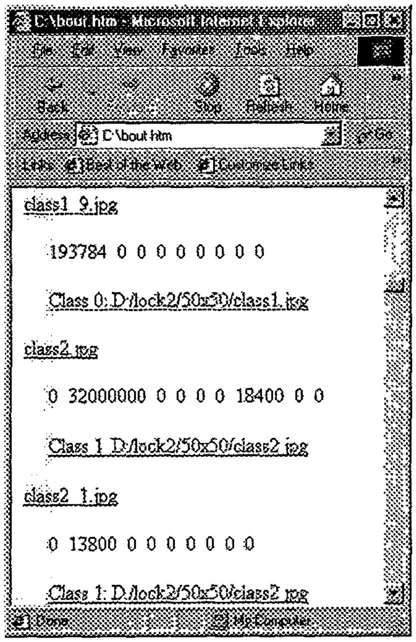

Figure 15. Sample Classification output of the Present Invention. The classification output is a list of triplets.